UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/659,975	09/11/2003	Kraig Allan Bottemiller	ROC920030180US1	5041

30206 7590 08/30/2007

IBM CORPORATION
ROCHESTER IP LAW DEPT. 917
3605 HIGHWAY 52 NORTH
ROCHESTER, MN 55901-7829

EXAMINER	
FENNEMA, ROBERT E	
ART UNIT	**PAPER NUMBER**
2183	

MAIL DATE	DELIVERY MODE
08/30/2007	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

5

Office Action Summary	Application No.	Applicant(s)	
	10/659,975	BOTTEMILLER ET AL.	
	Examiner	**Art Unit**	
	Robert E. Fennema	2183	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on 22 June 2007.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) 1,4 and 6-18 is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) 1, 4, and 6-18 is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☐ Notice of References Cited (PTO-892)

2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)

3) ☐ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date _____.

4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .

5) ☐ Notice of Informal Patent Application

6) ☐ Other: _____.

DETAILED ACTION

1. Claims 1, 4, and 6-18 have been considered. Claims 1, 4, 6, 9-12, 15, and 16 amended as per Applicant's request. Claims 2-3 and 5 cancelled as per Applicant's request.

Claim Rejections - 35 USC § 103

2. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

3. Claims 1, 4, 6, 8-13, and 15-17 are rejected under 35 U.S.C. 103(a) as being unpatentable over Swoboda et al. (USPN 6,643,803, herein Swoboda), in view of Torrey et al. (USPN 6,145,123, herein Torrey), further in view of Official Notice, further in view of Hoyle et al. (USPN 6,453,405, herein Hoyle).

4. As per Claim 1, Swoboda teaches: A method for implementing atomic data tracing in a processor system including an auxiliary processor unit (Column 4, Lines 52-62, the Test/Debug Host) coupled to a central processor unit (CPU) (Figure 1, Target device 10), using the auxiliary processor unit (APU) to perform the steps of:

identifying a trace instruction (Column 4, Lines 59-62);

said trace instruction including a primary op code (Column 2, Lines 15-19) and indicating General Purpose Registers (GPRs) containing information to identify a first

GPR containing data to be written into a current trace entry of a trace buffer (Column 4, Lines 59-62, what to trace, for example, memory 0x0 or 0x100), said data to be written being saved automatically in at least one GPR including said first GPR during normal context switch processing (this is what a context switch is, saving and restoring registers, and Examiner is taking Official Notice that one of ordinary skill in the art would have been motivated to use multithreading (thus context switching) in Swoboda's invention, as there are a multitude of advantages for doing so, such as being able to execute instructions on a long-latency event, or being able to execute multiple processes on a single processor);

signaling the CPU with a pipeline stall signal for stalling a CPU instruction stream pipeline (Column 26, Lines 46-48);

signaling the CPU with an op done signal for allowing the CPU to continue with instruction stream pipeline processing (Column 28, Lines 42-43), but fails to teach:

and to identify a set of trace engine registers defining a trace engine to use for said trace instruction; said trace engine including said trace buffer;

said trace buffer and said set of trace engine registers defining said trace engine being accessible by the APU;

checking for an enabled trace engine for said trace instruction,

writing trace data into a trace buffer responsive to an identified enabled trace engine for said trace instruction utilizing said set of trace engine registers defining said trace engine including writing trace data into said trace buffer from multiple execution contexts; and said trace engine including a set of device control registers (DCRs)

accessible by the APU and used to determine where to write the data into said trace buffer; said set of device control registers (DCRs) including a trace buffer pointer register storing a base address of said trace buffer and an offset indicating a current trace buffer entry, and a base address mask register storing a mask indicating which bits in said trace buffer pointer register hold said base address and which hold said offset, said base address mask register used to determine a wrap point of said trace buffer.

While Swoboda teaches a trace instruction, he does not specifically teach a trace engine, nor has it been taught that a check for an enabled trace engine. However, Swoboda does teach that debug data is sent to a test host through a Jtag port. However, Torrey teaches a multi-context system (Column 1, Lines 49-50) which gets debug information and outputs it to a system through a Jtag port, which uses a trace buffer in order to store the data to be sent out (Column 5, Lines 45-58) to account for the differences in speed between the I/O interface and the system clock. Torrey further teaches that this tracing can be enabled or disabled with a bit, such that the tracing does not need to occur when the user is not debugging the circuit (Column 5, Lines 25-27). Furthermore, it is required that some register in Torrey keep track of where to write to the trace buffer, as seen in Columns 5 and 6, specifically, Column 6, Lines 30-55 shows that reads and write can occur simultaneously, and to avoid overwriting the data, there must be some mechanism to indicate where to write, despite it not being specifically disclosed. Given the advantage Torrey provides, which is being able to store the trace data to be sent out to the debug host, since it can not be sent out at the same

speed at which it is processed, one of ordinary skill in the art at the time the invention was made would have been motivated to include a trace buffer (trace engine) such as Torrey's into Swoboda's invention, along with the other advantageous features disclosed above.

While Swoboda and Torrey disclose a system to trace execution using a trace buffer (requiring addressing provided by the DCRs), the specific method of addressing has not been taught in either reference. However, Hoyle teaches a method to address a circular buffer (which is how the trace buffer in Torrey is set up as), involving a base address, and offset, and a mask (These features are taught in several locations in the reference, but can be seen together in Hoyle's Claim 11, which will be referred to for simplicities sake, but Column 3, Lines 11-14 could also be seen). Given the need to address the trace buffer in some way, one of ordinary skill in the art would have been motivated to use a method such as the one disclosed by Hoyle, in able to properly address and use the trace buffer as disclosed by Torrey, and as stated in the previous rejection, the system of Swoboda would have needed to include some indication of this information as well to ensure the data was written to the correct location. Therefore, given this need to address the buffer, one of ordinary skill in the art at the time the invention was made would have been motivated to include the teachings of Hoyle in the combination of Swoboda and Torrey.

5. As per Claim 4, Hoyle teaches: The method for implementing atomic data tracing as recited in claim 1 wherein the step of writing trace data into said trace buffer includes

updating said offset into said current trace buffer entry of said trace buffer pointer register (Claim 11, and Column 3, Lines 11-14).

6. As per Claim 6, Torrey teaches: The method for implementing atomic data tracing as recited in claim 1 wherein said set of device control registers (DCRS) include a control register storing an enabled bit indicating whether or not said trace engine for said trace instruction is enabled, said enabled bit being used for allowing data tracing to be turned on and off on the fly (Column 5, Lines 25-28).

7. As per Claim 8, Torrey teaches: The method for implementing atomic data tracing as recited in claim 6, wherein said control register includes a number field indicating a number of bytes to be traced (Column 8, Lines 29-31); and wherein the step of writing trace data into said trace buffer includes updating said offset into said current trace buffer entry of said trace buffer pointer register by said number of bytes (For multiple writes with a larger-than one length trace, the offset would be required to update to continue to point to a valid area in the buffer to write to).

8. As per Claim 9, Torrey teaches: The method for implementing atomic data tracing as recited in claim 1 wherein said trace instruction includes a number field indicating a number of bytes to be traced (Column 8, Lines 29-31); and wherein the step of writing trace data into said trace buffer includes updating said offset into said current trace buffer entry of said trace buffer pointer register by said number of bytes (For

multiple writes with a larger-than one length trace, the offset would be required to update to continue to point to a valid area in the buffer to write to).

9. As per Claim 10, Swoboda teaches: The method for implementing atomic data tracing as recited in claim 1 responsive to identifying no enabled trace engine for said trace instruction, signaling the CPU with said op done signal for allowing the CPU to continue with instruction processing without writing trace data (Column 28, Lines 42-43).

10. As per Claim 11, Swoboda teaches: An apparatus for implementing atomic data tracing in a processor system including an auxiliary processor unit (APU) (Column 4, Lines 52-62, the Test/Debug Host) coupled to a central processor unit (CPU) (Figure 1, Target device 10), said apparatus comprising:

a trace instruction (Column 4, Lines 59-62); said trace instruction including a primary op code (Column 2, Lines 15-19) and said second GPR indicating a first GPR containing data to be written into a current trace entry in said trace buffer (Column 4, Lines 59-62, what to trace, for example, memory 0x0 or 0x100);

said data to be written being saved automatically in at least one GPR including said first GPR during normal context switch processing (this is what a context switch is, saving and restoring registers, and Examiner is taking Official Notice that one of ordinary skill in the art would have been motivated to use multithreading (thus context switching) in Swoboda's invention, as there are a multitude of advantages for doing so,

such as being able to execute instructions on a long-latency event, or being able to execute multiple processes on a single processor);

the APU processes said trace instruction performing the steps of signaling the CPU with a pipeline stall signal for stalling a CPU instruction stream pipeline (Column 26, Lines 46-48); and signaling the CPU with an op done signal for allowing the CPU to continue with instruction stream pipeline processing (Column 28, Lines 42-43), but fails to teach:

a trace engine; said trace engine including a set of device control registers (DCRs) accessible by the APU, and a trace buffer; said set of device control registers (DCRs) including a trace buffer pointer register storing a base address of said trace buffer and an offset indicating a current trace buffer entry and including a base address mask register storing a mask indicating which bits in said trace buffer pointer register hold said base address and which hold said offset; said base address mask register used to determine a wrap point of said trace buffer;

said trace instruction including encoded first and second general purpose registers (GPRs), said first GPR containing an index to said trace engine DCRs,

responsive to identifying an enabled trace engine for said trace instruction, and

writing trace data into said trace buffer utilizing said set of device control registers (DCRs) included in said trace engine to determine where to write the data into said trace buffer and including writing trace data into said trace buffer from multiple execution contexts.

While Swoboda teaches a trace instruction, he does not specifically teach a trace engine, nor has it been taught that a check for an enabled trace engine. However, Swoboda does teach that debug data is sent to a test host through a Jtag port. However, Torrey teaches a multi-context system (Column 1, Lines 49-50) which gets debug information and outputs it to a system through a Jtag port, which uses a trace buffer in order to store the data to be sent out (Column 5, Lines 45-58) to account for the differences in speed between the I/O interface and the system clock. Torrey further teaches that this tracing can be enabled or disabled with a bit, such that the tracing does not need to occur when the user is not debugging the circuit (Column 5, Lines 25-27). Furthermore, it is required that some register in Torrey keep track of where to write to the trace buffer, as seen in Columns 5 and 6, specifically, Column 6, Lines 30-55 shows that reads and write can occur simultaneously, and to avoid overwriting the data, there must be some mechanism to indicate where to write, despite it not being specifically disclosed. Given the advantage Torrey provides, which is being able to store the trace data to be sent out to the debug host, since it can not be sent out at the same speed at which it is processed, one of ordinary skill in the art at the time the invention was made would have been motivated to include a trace buffer (trace engine) such as Torrey's into Swoboda's invention, along with the other advantageous features disclosed above.

While Swoboda and Torrey disclose a system to trace execution using a trace buffer (requiring addressing provided by the DCRs), the specific method of addressing has not been taught in either reference. However, Hoyle teaches a method to address a

circular buffer (which is how the trace buffer in Torrey is set up as), involving a base address, and offset, and a mask (These features are taught in several locations in the reference, but can be seen together in Hoyle's Claim 11, which will be referred to for simplicities sake, but Column 3, Lines 11-14 could also be seen). Given the need to address the trace buffer in some way, one of ordinary skill in the art would have been motivated to use a method such as the one disclosed by Hoyle, in able to properly address and use the trace buffer as disclosed by Torrey, and as stated in the previous rejection, the system of Swoboda would have needed to include some indication of this information as well to ensure the data was written to the correct location. Therefore, given this need to address the buffer, one of ordinary skill in the art at the time the invention was made would have been motivated to include the teachings of Hoyle in the combination of Swoboda and Torrey.

11. Claim 15 is substantially similar to Claim 11, and is rejected under 35 U.S.C. 103(a) for the same reasons.

12. As per Claim 12, Swoboda and Torrey teach: An apparatus for implementing atomic data tracing in a processor system as recited in claim 11,

a control register storing an enabled bit indicating whether or not said trace engine for said trace instruction is enabled, said enabled bit being used for allowing data tracing to be turned on and off on the fly (Torrey, Column 5, Lines 25-28), but fail to teach:

wherein said set of device control registers (DCRs).

While Swoboda and Torrey disclose a system to trace execution using a trace buffer (requiring addressing provided by the DCRs), the specific method of addressing has not been taught in either reference. However, Hoyle teaches a method to address a circular buffer (which is how the trace buffer in Torrey is set up as), involving a base address, and offset, and a mask (These features are taught in several locations in the reference, but can be seen together in Hoyle's Claim 11, which will be referred to for simplicities sake). Given the need to address the trace buffer in some way, one of ordinary skill in the art would have been motivated to use a method such as the one disclosed by Hoyle, in able to properly address and use the trace buffer as disclosed by Torrey, and as stated in the previous rejection, the system of Swoboda would have needed to include some indication of this information as well to ensure the data was written to the correct location. Therefore, given this need to address the buffer, one of ordinary skill in the art at the time the invention was made would have been motivated to include the teachings of Hoyle in the combination of Swoboda and Torrey.

13. As per Claim 13, Hoyle teaches: An apparatus for implementing atomic data tracing in a processor system as recited in claim 12 wherein the APU updates said offset to said current trace buffer entry of said trace buffer pointer register for each trace data entry written to said trace buffer (Column 16, Lines 35-58, the offset is aligned to the proper location for each write).

14. Claims 7, 14 and 18 are rejected under 35 U.S.C. 103(a) as being unpatentable over Swoboda, Torrey, and Hoyle, further in view of DeAngelis et al. (USPN 5,226,153, herein DeAngelis).

15. As per Claim 7, Swoboda teaches: The method for implementing atomic data tracing as recited in claim 6, but fails to teach:

wherein said control register includes a time stamp value indicating whether or not a time stamp should be traced; and wherein the step of writing trace data into said trace buffer includes writing a time stamp with said trace data responsive to said control register time stamp value.

While Swoboda teaches the Apparatus as disclosed in the previous claims, Swoboda and the combinations disclosed above remain silent to writing a time stamp with the trace data. However, DeAngelis discloses that a recurring problem in data traces is the inability to relate the data traces to each other in time (Column 1, Lines 37-43). DeAngelis discloses a method to insert a time stamp into traces to remedy this problem (Column 13, Lines 10-28). Furthermore, given that Torrey taught a control register which enabled or disabled traces, the same bits which controlled the trace being enabled would also enable or disable the time stamps as disclosed by DeAngelis. Given the problem of being unable to correlate data, and DeAngelis' solution of using a time stamp in order to overcome this problem, one of ordinary skill in the art at the time the invention was made would have been motivated to include a time stamp in the invention of Swoboda, Torrey, and Hoyle, in order to overcome this limitation.

16. As per Claim 14, Swoboda teaches: An apparatus for implementing atomic data tracing in a processor system as recited in claim 12, but fails to teach:

wherein said control register includes a time stamp value indicating whether or not a time stamp should be traced; and the APU writes a time stamp with said trace data responsive to said control register time stamp value.

While Swoboda teaches the Apparatus as disclosed in the previous claims, Swoboda and the combinations disclosed above remain silent to writing a time stamp with the trace data. However, DeAngelis discloses that a recurring problem in data traces is the inability to relate the data traces to each other in time (Column 1, Lines 37-43). DeAngelis discloses a method to insert a time stamp into traces to remedy this problem (Column 13, Lines 10-28). Furthermore, given that Torrey taught a control register which enabled or disabled traces, the same bits which controlled the trace being enabled would also enable or disable the time stamps as disclosed by DeAngelis. Given the problem of being unable to correlate data, and DeAngelis' solution of using a time stamp in order to overcome this problem, one of ordinary skill in the art at the time the invention was made would have been motivated to include a time stamp in the invention of Swoboda, Torrey, and Hoyle, in order to overcome this limitation.

17. Claims 16-18 are substantially similar to Claims 12-14, and have been rejected under 35 U.S.C. 103(a) for the same reasons.

Response to Arguments

18. Examiner appreciates Applicant's attempt to amend the claims to further define the invention over the prior art. However, Examiner has not been completely convinced that the claims currently overcome the art, and Examiner will address the Applicant's arguments to explain why.

Applicant has argued on Pages 16-17 of the remarks that each of the independent claims disclose writing data to a single trace buffer, which holds data from multiple contexts. While Examiner will admit that Torrey (or any of the other references) do not explicitly come out and indicate they have this feature, Examiner believes it is implied in the references in a strong enough manner to say that the references do this. Torrey teaches in Column 1 that the typical computer of which he has improved is a multitasking/multithreading machine, something which is fairly typical in the art. Torrey further teaches in Columns 5 and 6 that the tracing is done in a trace unit, with a single buffer, which outputs data to the Jtag ports as a host system requests it. Examiner believes that because there is only mention of a single trace unit and trace buffer, and that it is used by making use of a breakpoint register, that all the contexts being used in Torrey's system use the same trace buffer to export information to the host system, thus Examiner believes there is enough evidence to say that Torrey has a trace buffer in which multiple contexts write to, as opposed to a system in which there may be multiple trace buffers, one for each context. For these reasons, the Examiner believes that this limitation to the claims does not currently overcome the art.

The Applicant has further argued that the trace engine, and the step of defining the trace buffer and registers to create a trace engine is not taught in the cited art. Examiner believes that the cited art does teach the trace engine, as defined by a set of control registers and a buffer, as shown in the previous and current rejection, the combination of references teaches these features. As for the step of defining, Examiner believes that this may be an issue that upon further clarification, may help to overcome the art, but currently is not able to do so. As Examiner has interpreted this limitation in the independent claims, it appears that the "defining" step is just a way of saying what the buffer is, for example, if the system knows that the buffer and a certain number of registers is the trace unit, then it has been "defined" as the trace engine. If there is something more to this step of defining other than just the system knowing what the trace engine is, then that may be something which could overcome the art, however, Examiner does not feel that anything more than that is currently expressed in the claims, although if Applicant disagrees, Applicant is encouraged to make their case, however the Examiner currently believes it is too broad to actually mean anything else.

Additionally, on pages 18 and 19, Applicant has argued that Torrey fails to teach that the instruction would be required to specify where in the buffer to write, as Examiner has said must be done in order to use it. Examiner believes he either misinterpreted the claims earlier, or was not clear about getting his rejection across for this limitation, and regrets any confusion, so has clarified the rejection, and will explain it here, as Examiner still believes this feature is taught in Torrey, although not in the manner that Examiner originally detailed. Examiner admits that the instruction itself

does not specific where in the trace buffer to write, however, a look at the claims shows that the instruction is not claimed to do this, but rather, that one of the control registers indicates where to write the data. Now, the point that Examiner was trying to get across is that Torrey teaches that a trace buffer holds any information which cannot be immediately sent out to the host unit, and that the trace buffer may be written with trace data at the same time that data is being read out of the trace buffer to be sent to the host system (See columns 5 and 6, which discuss this in detail). Now, what the Examiner was trying to originally get across is that despite Torrey not mentioning specifically how the trace buffer is addressed to be written into (at least that the Examiner could find), in order to write to the buffer, it is necessary that some address to the buffer is held somewhere, so that it can be written, as data cannot just be randomly written to the buffer, and even if it was, that random value would still have to come from somewhere. So by this logic, Examiner was attempting to get across the point that some register must be storing some value in Torrey's invention to tell the processor where in the trace buffer to write the trace data, and Examiner considers that register, whatever it is, to be part of the trace control registers.

Lastly, the Applicant has argued that the references used to rejection the claims fail to provide a suggestion or motivation to modify the reference or combine the teachings. Examiner believes that the motivation does exist to combine these references, and refers to the rejections themselves, where this motivation has been laid out in more detail. However, Examiner also refers to the recent Supreme Court ruling of *KSR International Co. v Teleflex Inc.,* which has essentially stated that if a person of

ordinary skill in the art can implement a predicable variation, it is proper to combine under 103. In this case, the rejections have been combined in a way that implement a known technique in a known device to yield predictable results. Both Torrey, Hoyle, and DeAngelis describe improvements which yield predictable and beneficial results in analogous art to Swoboda, and as a result, the motivation exists to combine the references.

If Applicant wishes, Examiner encourages Applicant to contact the Examiner, and the Examiner would be willing to try to discuss any possible changes that may help overcome the art through either further defining the invention, or helping to clarify some of the issues addressed above, or to discuss the rejection in an attempt to bring the case to allowability.

Conclusion

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Robert E. Fennema whose telephone number is (571) 272-2748. The examiner can normally be reached on Monday-Friday, 8:45-6:15.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Eddie Chan can be reached on (571) 272-4162. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

Robert E Fennema
Examiner
Art Unit 2183

RF

EDDIE CHAN
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100